Big Data Analytics

Lecture 1: Introduction and Motivation

Huanle Xu

Motivation

- Do you want to work in these companies?

Motivation of the Course

- Do you want to understand what is big data? What are the main characteristics of big data?
- Do you want to understand the infrastructure and techniques of big data analytics?
- Do you want to know the research challenges in the area of big data learning and mining?

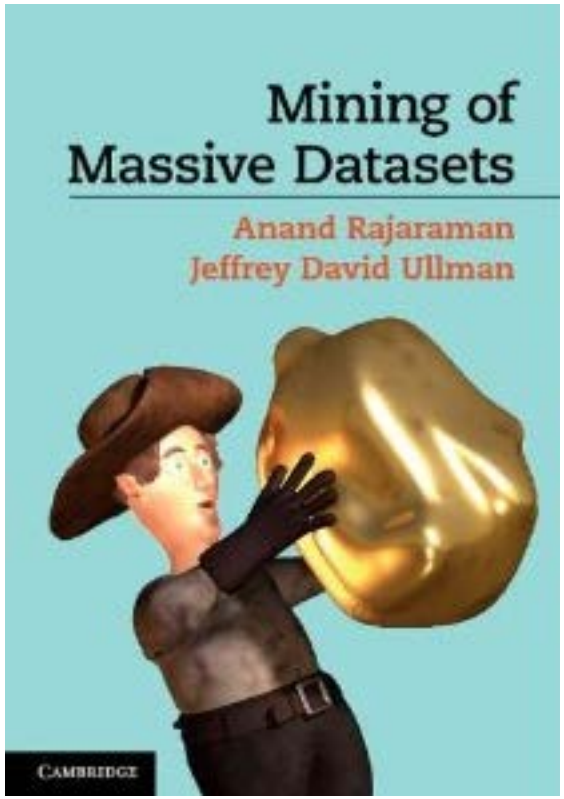
Course Objective

1. To understand the current key issues on big data and the associated business/scientific data applications;
2. To teach the fundamental techniques and frameworks in achieving big data analytics with scalability and streaming capability
3. To understand basic optimization methods towards solving big data related problems
4. Able to apply software tools for big data analytics

Course Description

- Course Homepage: https://xuhappy.github.io/courses/BigData/
- This course aims at teaching students the state-of-the-art big data analytics, including techniques, software, applications, and perspectives with massive data.
- The class will cover, but not be limited to, the following topics:
  - cloud computing, big data processing frameworks, distributed file systems such as Google File System, Hadoop Distributed File System, CloudStore, and map-reduce technology;
  - machine learning technology, SVM models, Deep Neural Networks
  - Data Mining Methods, Clustering, Dimension Reduction, Recommendation systems
  - optimization methods, convex optimization, online learning

Textbook for Reference



- Mining of Massive Datasets
- Anand Rajaraman
  - web and technology entrepreneur
  - co-founder of Cambrian Ventures and Kosmix
  - co-founder of Junglee Corp (acquired by Amazon for a retail platform)
- Jeff Ullman
  - The Stanford W. Ascherman Professor of Computer Science (Emeritus)
  - Interests in database theory, database integration, data mining, and education using the information infrastructure.

Textbook

- Amazon
  - http://www.amazon.com/Mining-Massive-Datasets-Anand-Rajaraman/dp/1107015359
- PDF of the book for online viewing
  - http://infolab.stanford.edu/~ullman/mmds.html

Prerequisites

- Algorithms
  - Basic data structures
- Operating Systems
  - Linux
- Basic mathematics
  - Moments, typical distributions, …
- Programming
  - Your choice
- **We provide some background, but the class will be fast paced**

What Will We Learn?

- We will learn to analyze different types of data:
  - Data is high dimensional
  - Data is a graph
  - Data is infinite/never-ending
  - Data is labeled
- We will learn to use different models of computation:
  - MapReduce
  - Streams and online algorithms
  - Single machine in-memory

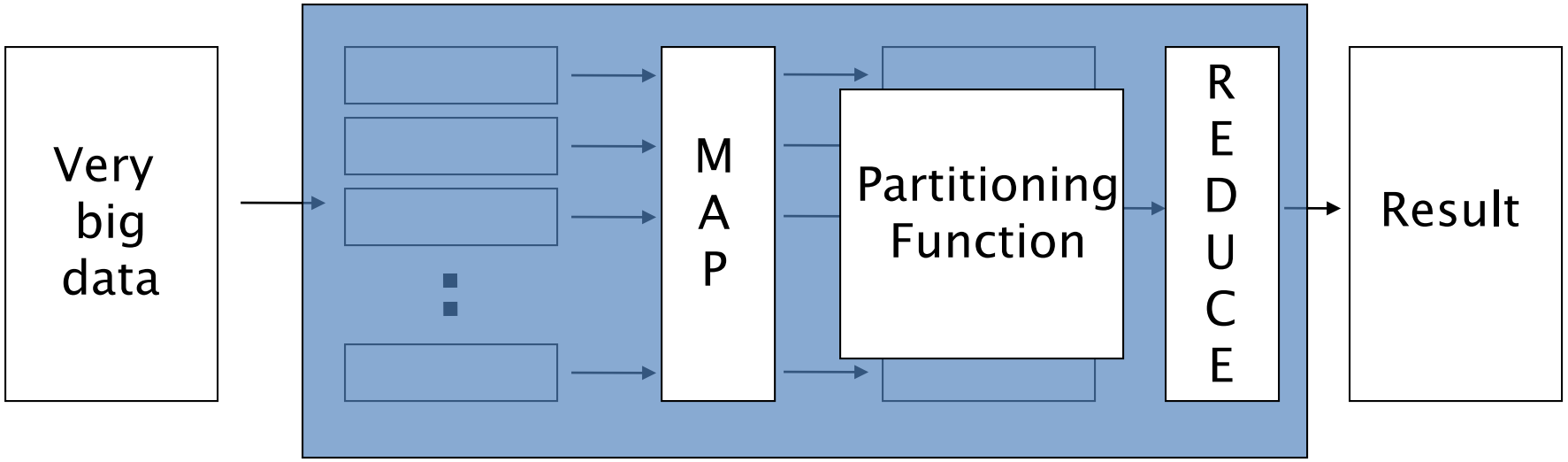
What Will We Learn?

- We will learn to solve real-world problems:
  - Recommender systems
  - Link analysis
  - Digit handwritten recognition
  - Community detection
- We will learn various "tools":
  - Linear algebra (SVD, Rec. Sys., Communities)
  - Optimization (stochastic gradient descent)
  - Various big data processing frameworks: MapReduce, Hadoop

Class Project

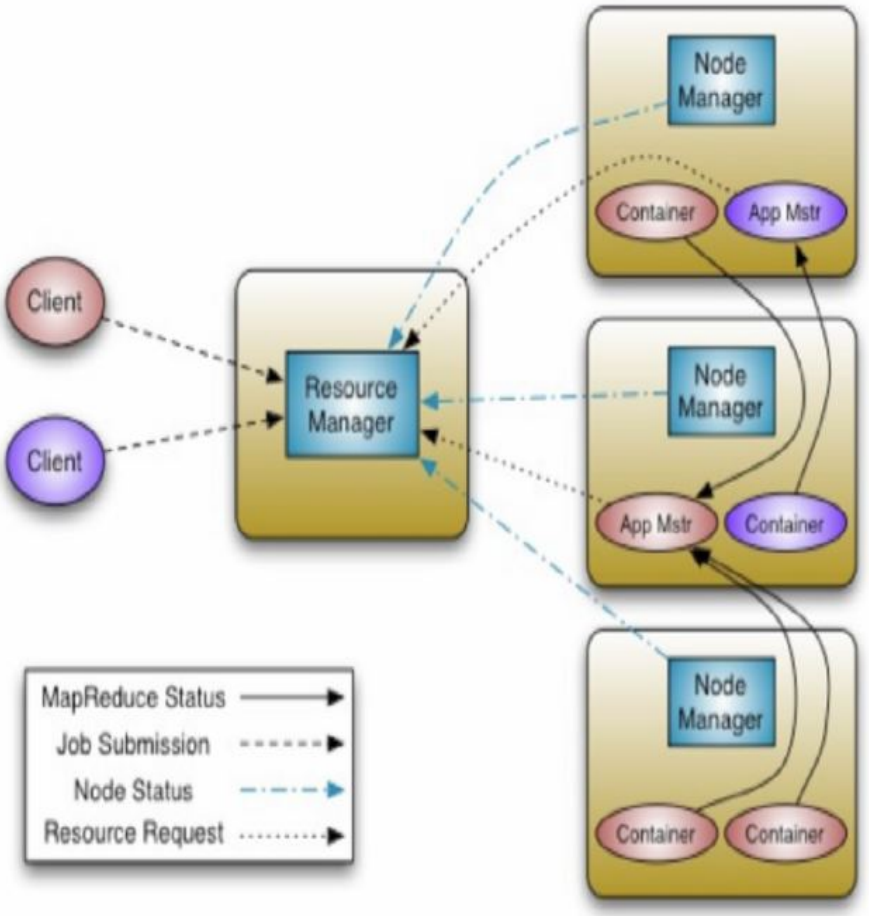
- Project is for everyone
- One-two person/s per project group
- Each group is to design and implement a big data-related project of choice
- Detailed schedule will be announced later

MapReduce



- Map:
  - Accepts *input* key/value pair
  - Emits *intermediate* key/value pair
- Reduce:
  - Accepts *intermediate* key/value* pair
  - Emits *output* key/value pair

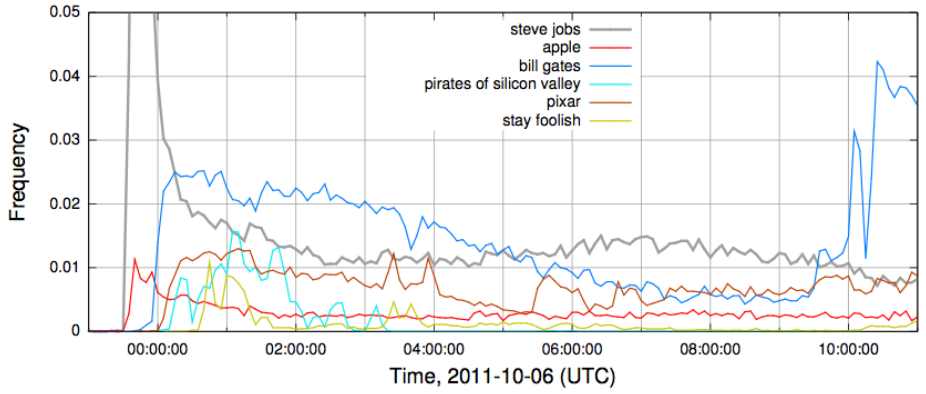
YARN Architectural Overview

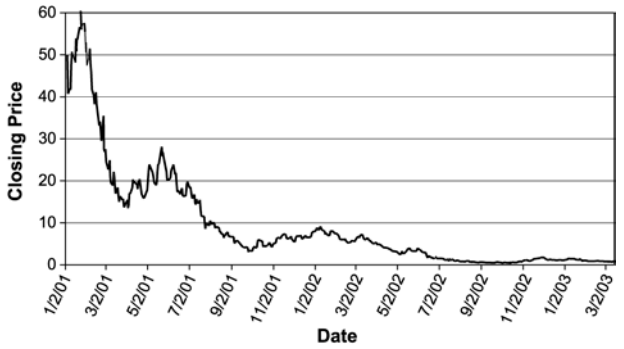


- Scalability - Clusters of 6,000-10,000 machines
  - Each machine with 16 cores, 48G/96G RAM, 24TB/36TB disks
  - 100,000+ concurrent tasks
  - 10,000 concurrent jobs

Mining Data Stream

- Stream Management is important when the input rate is controlled externally:
  - Google queries
  - Twitter or Facebook status updates
- We can think of the data as infinite and non-stationary (the distribution changes over time)

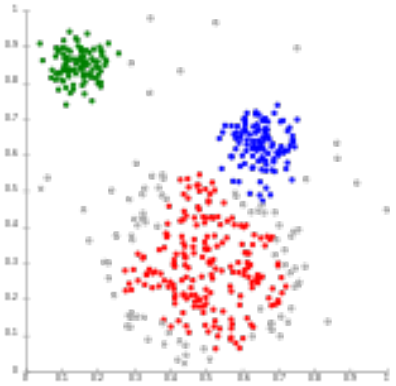


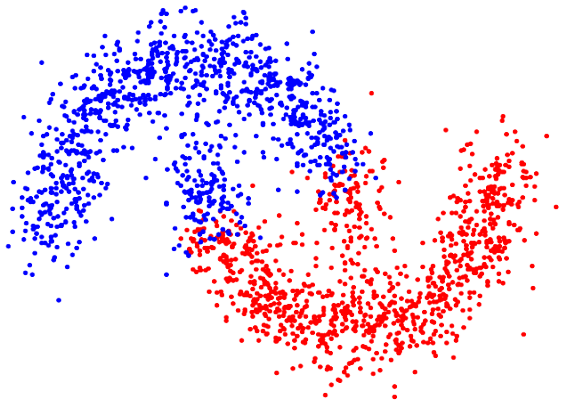


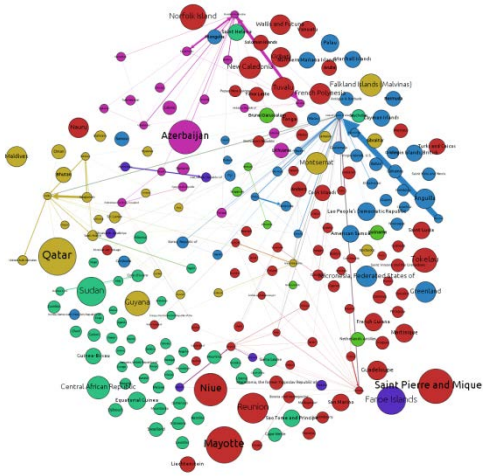
Note: I2 stock price from Jan 2001 to March 2003

Clustering

- Given a set of points, with a notion of distance between points, group the points into some number of clusters, so that
  - Members of a cluster are close/similar to each other
  - Members of different clusters are dissimilar

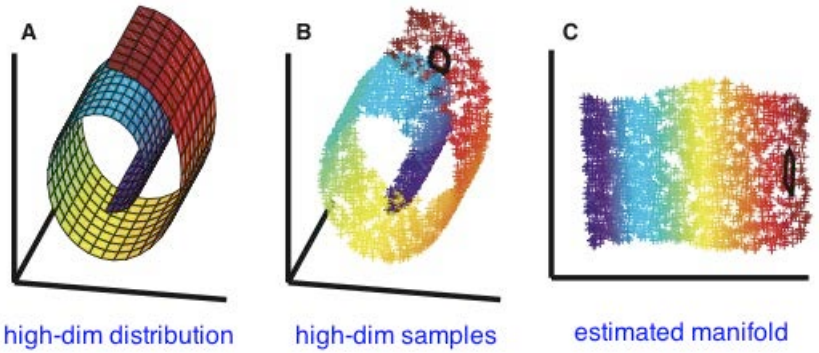


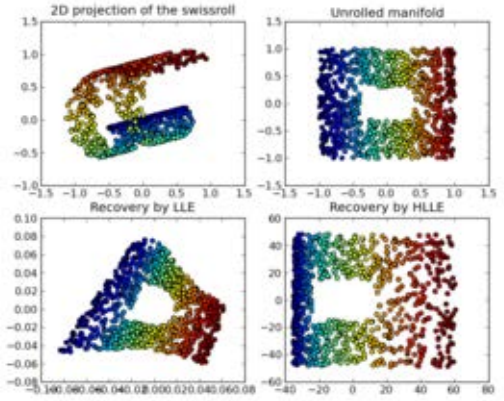




Dimensionality Reduction

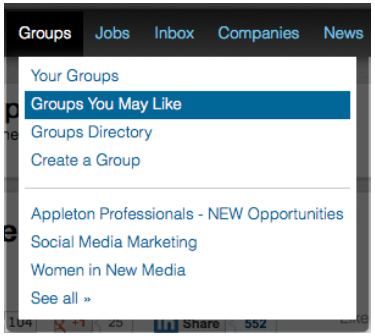
- Discover hidden correlations/topics
  - Words that occur commonly together
- Remove redundant and noisy features
  - Not all words are useful
- Interpretation and visualization
- Easier storage and processing of the data





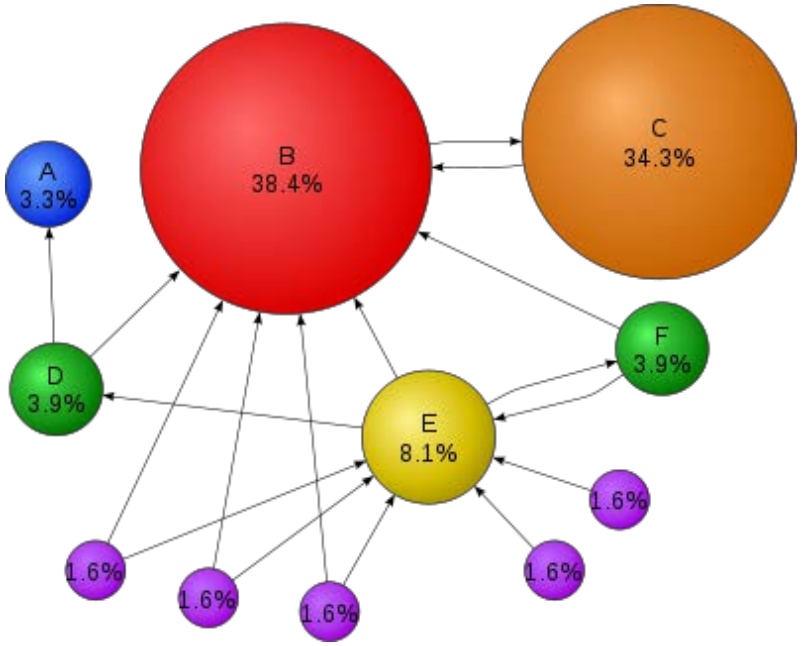
Recommender System

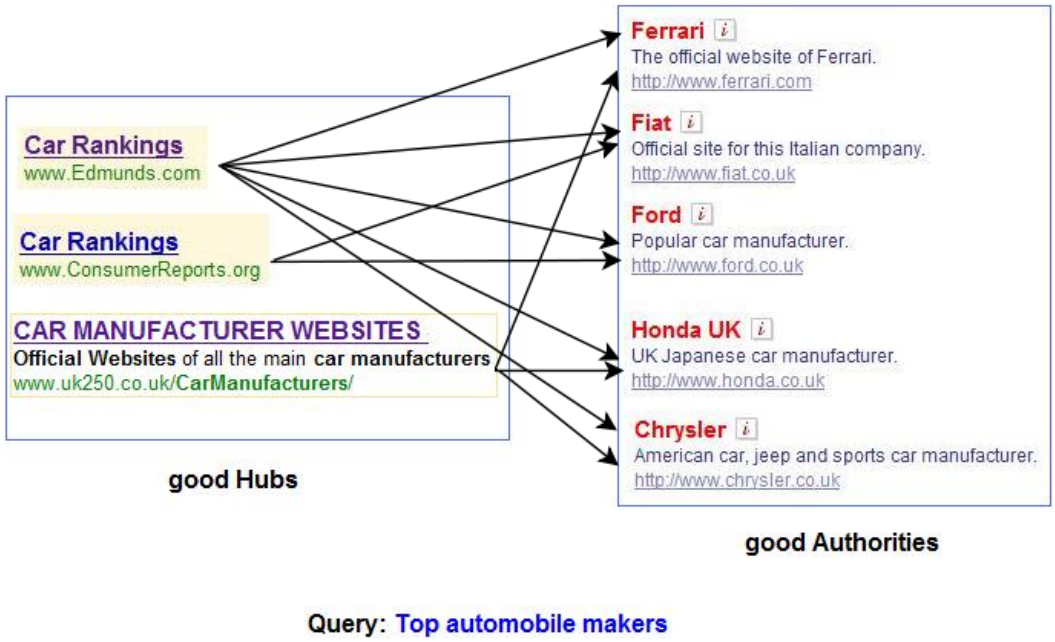
- Main idea: Recommend items to customer ***x*** similar to previous items rated highly by ***x***
- Example:
  - Movie recommendations
    - Recommend movies with same actor(s), director, genre, …
  - Websites, blogs, news
    - Recommend other sites with "similar" content



Link Analysis

- Computing importance of nodes in a graph



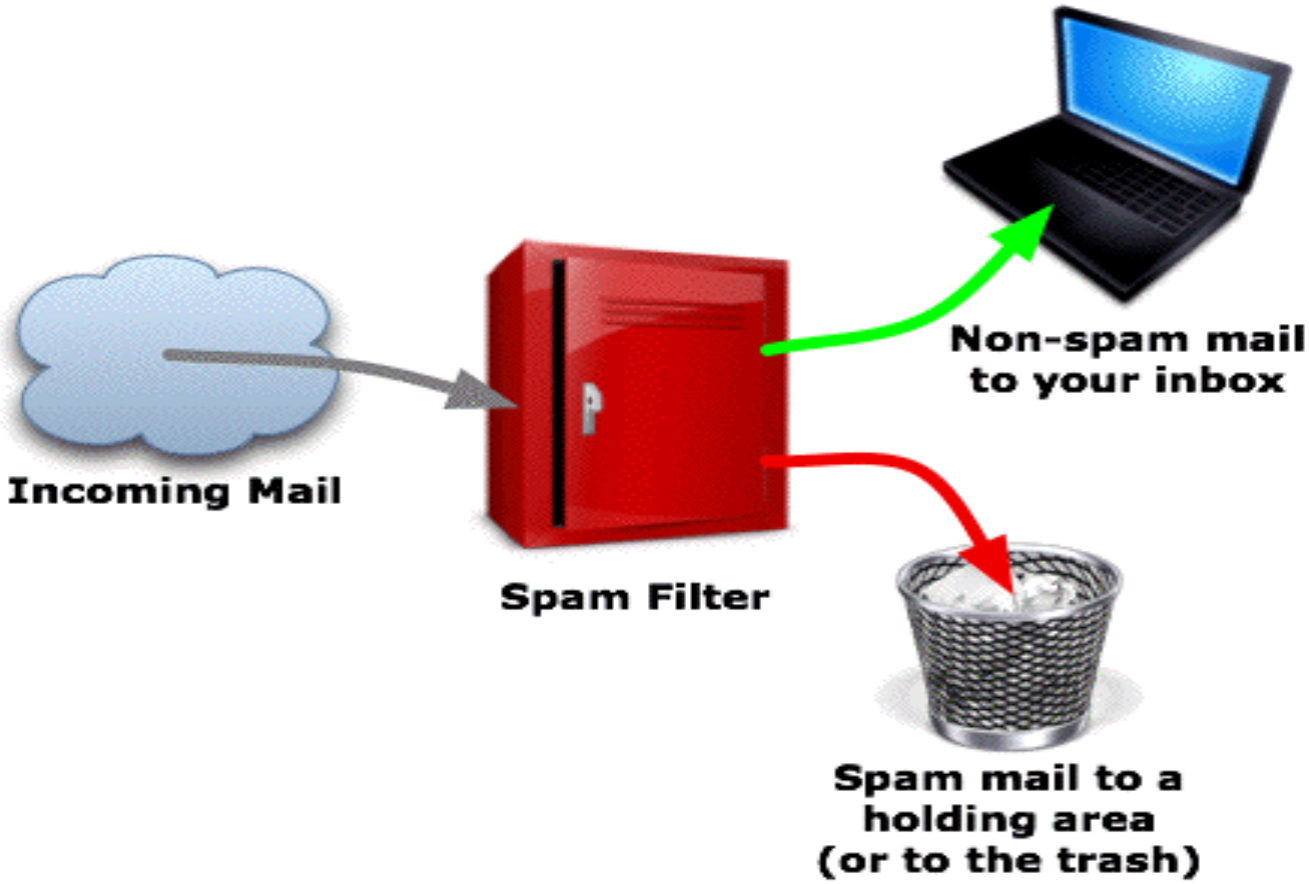


Query: Top automobile makers

Large Scale Classification

How does a computer know whether a news is technology and health? Classification

Online Learning Algorithms



How to update the decision function and make decision as a new sample comes?

Introduction to Big Data

Definition of Big Data

- Big data is a collection of data sets so large and complex that it becomes difficult to process using on-hand database management tools or traditional data processing applications.

From wiki

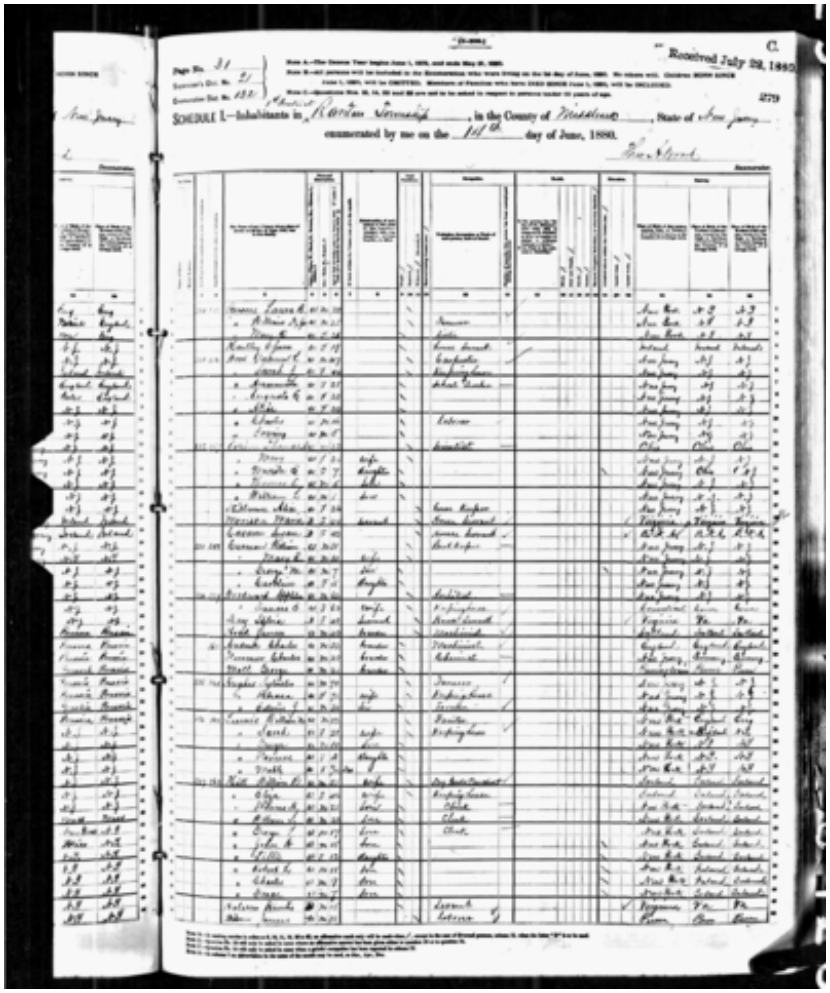
Evolution of Big Data

- Birth: 1880 US census
- Adolescence: Big Science
- Modern Era: Big Business

Birth: 1880 US census

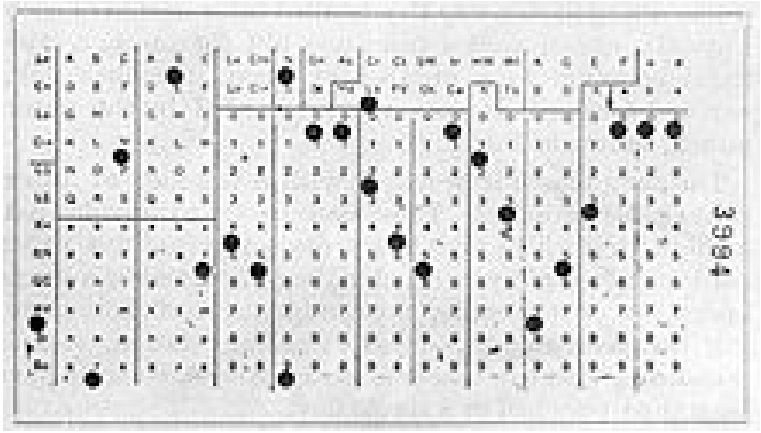
The First Big Data Challenge

- 1880 census
- 50 million people
- Age, gender (sex), occupation, education level, no. of insane people in household



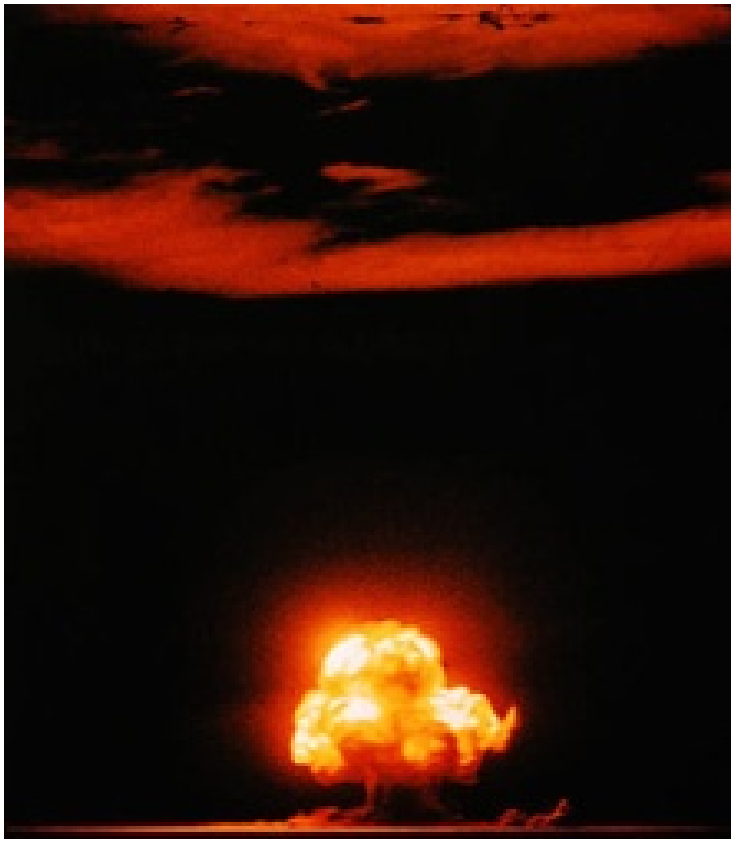
The First Big Data Solution

- Hollerith Tabulating System
- Punched cards – 80 variables
- Used for 1890 census
- 6 weeks instead of 7+ years



Manhattan Project (1946 - 1949)

- $2 billion (approx. 26 billion in 2013)
- Catalyst for “Big Science”



Space Program (1960s)

- Began in late 1950s
- An active area of big data nowadays

Big Science vs. Big Business

- Common
  - Need technologies to work with data
  - Use algorithms to mine data
- Big Science
  - Source: experiments and research conducted in controlled environments
  - Goals: to answer questions, or prove theories
- Big Business
  - Source: transactions in nature and little control
  - Goals: to discover new opportunities, measure efficiencies, uncover relationships

Big Data is Everywhere!

- Lots of data is being collected and warehoused
  - Web data, e-commerce
  - Purchases at department/ grocery stores
  - Bank/Credit Card transactions
  - Social Networks

How Big is Big ?

- 2008: Google processes 20 PetaByte per Day (Peta=10^{15})
- Apr 2009: Facebook has 2.5 PB user data + 15 TB/day
- May 2009: eBay has 6.5 PB user data + 50 TB/day
- 2011: Yahoo! Has 180-200 PB of data
- 2012: Facebook ingests 500TB/day

How many users and objects?

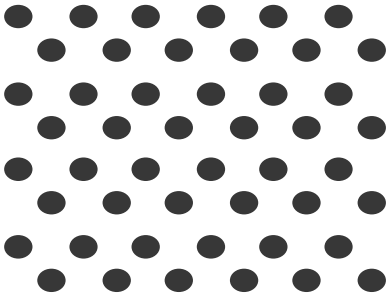
- Flickr has >6 billion photos
- Facebook has 1.15 billion active users
- Google is serving >1.2 billion queries/day on more than 27 billion items
- >2 billion videos/day watched on YouTube

How much data?

- Modern applications use massive data:
  - Rendering 'Avatar' movie required >1 petabyte of storage
  - eBay has >6.5 petabytes of user data
  - CERN's LHC will produce about 15 petabytes of data per year
  - In 2008, Google processed 20 petabytes per day
  - German Climate computing center dimensioned for 60 petabytes of climate data
  - Someone estimated in 2013 that Google had 10 exabytes on disk and ~ 5 exabytes on tape backup
  - NSA Utah Data Center is said to have 5 zettabyte (!)
- How much is a zettabyte?
  - 1,000,000,000,000,000,000,000 bytes
  - A stack of 1TB hard disks that is 25,400 km high

Characteristics of Big Data: 4V

Volume

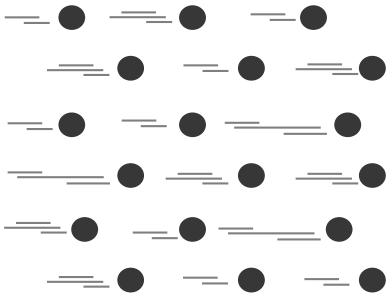


From terabytes to exabyte to zetabytes of existing data to process

IDC Analyze the Future

8 billion TB in 2015,
40 ZB in 2020
5.2TB per person

Velocity

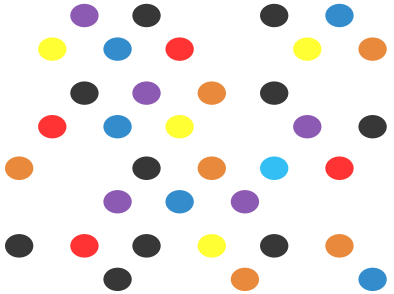


Batch data, real-time data, streaming data, milliseconds to seconds to respond

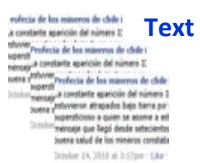
facebook

New sharing over 2.5 billion per day
new data over 500TB per day

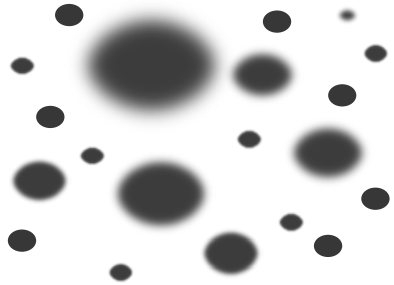
Variety



Structured, semi-structured, unstructured, text, pictures, multimedia



Veracity



Uncertainty due to data inconsistency & incompleteness, ambiguities, deception, model approximation

芦山地震十大不实谣言

2013年04月24日17:41　来源：人民网　手机看新闻

8.地震局内部消息成都9.2级地震

谣言：自称地震局内部人员的网民称发生9.2级地震。”

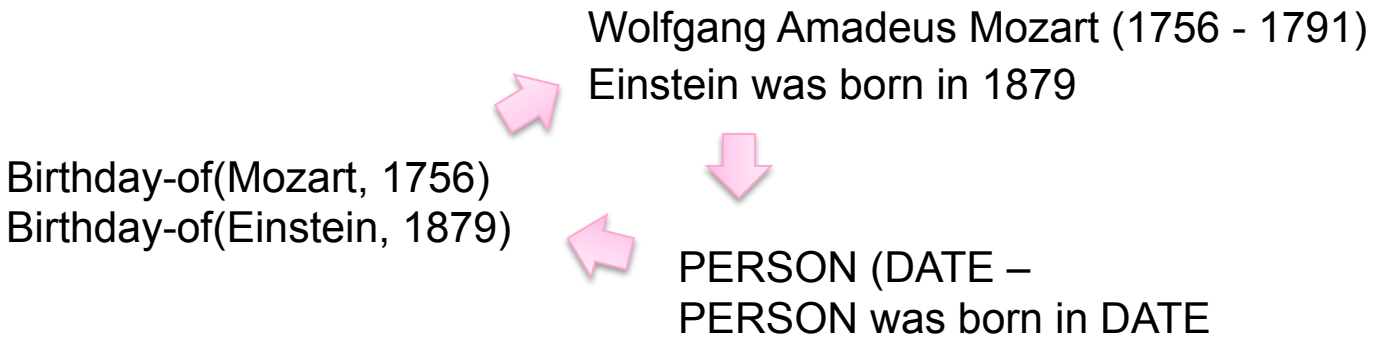
How much computation?

- No single computer can process that much data
  - Need many computers!
- How many computers do modern services need?
  - Facebook is thought to have more than 60,000 servers
  - 1&1 Internet has over 70,000 servers
  - Akamai has 95,000 servers in 71 countries
  - Intel has ~100,000 servers in 97 data centers
  - Microsoft reportedly had at least 200,000 servers in 2008
  - Google is thought to have more than 1 million servers, is planning for 10 million (according to Jeff Dean)

What to do with More Data ?

- Answering factoid questions
  - Pattern matching on the Web
  - Works amazingly well

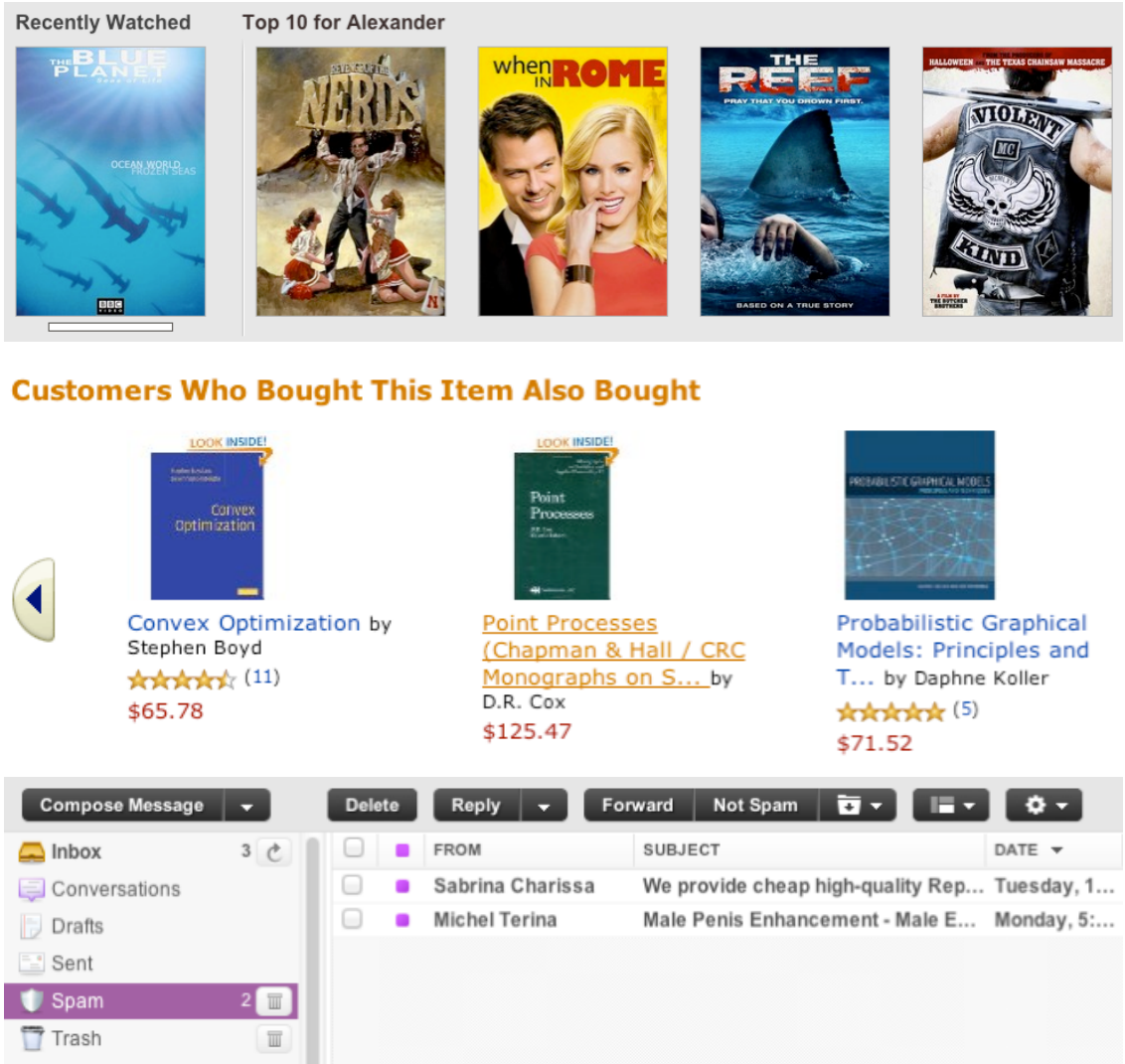
    Who shot Abraham Lincoln? --> ??? shot Abraham Lincoln
- Learning relations
  - Start with seed instances
  - Search for patterns on the Web
  - Using patterns to find more instances



(Brill et al., TREC 2001; Lin, ACM TOIS 2007)
(Agichtein and Gravano, DL 2000; Ravichandran and Hovy, ACL 2002; …)

What to do with More Data ? (cont'd)
Personalization

- 100-1000M users
  - Spam filtering
  - Personalized targeting & collaborative filtering
  - News recommendation
  - Advertising



Big Data Analytics

- Definition: A process of inspecting, cleaning, transforming, and modeling big data with the goal of discovering useful information, suggesting conclusions, and supporting decision making
- Hot in both industrial and research societies

Big Data Analytics

- Related conferences
  - IEEE Big Data
  - IEEE Big Data and Distributed Systems
  - WWW
  - KDD
  - WSDM
  - CIKM
  - SIGIR
  - AAAI/IJCAI
  - NIPS
  - ICML
  - TREC
  - ACL
  - EMNLP
  - COLING
  - …

Types of Analytics at eBay

- Basically measure anything possible - A **few** examples:

What is Data Mining?

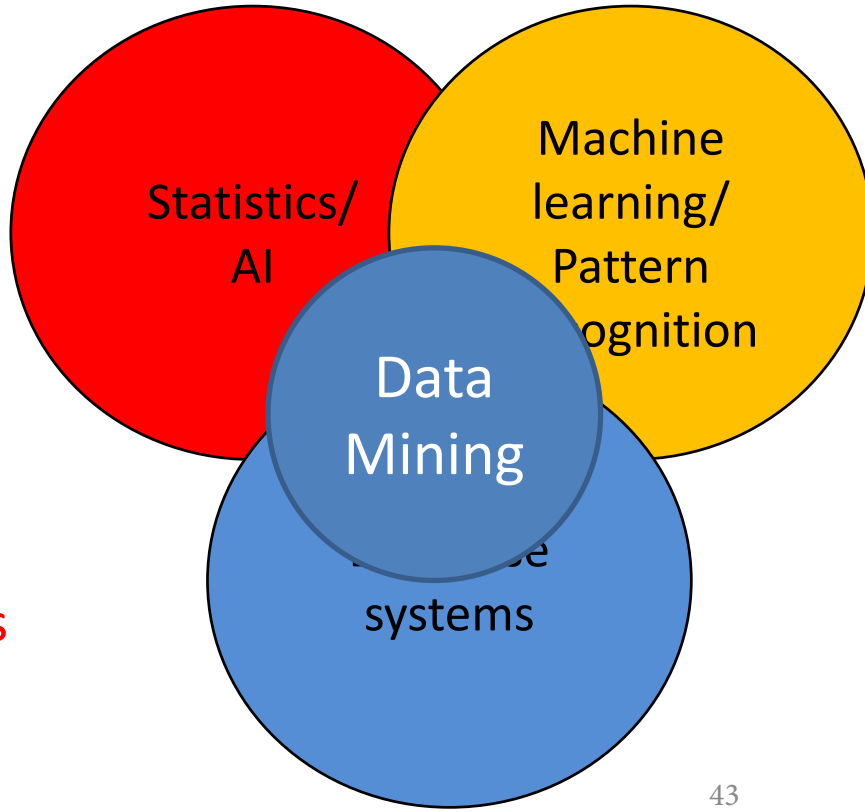
- Discovery of patterns and models that are:
  - Valid: hold on new data with some certainty
  - Useful: should be possible to act on the item
  - Unexpected: non-obvious to the system
  - Understandable: humans should be able to interpret the pattern
- A particular data analytic technique

Data Mining Tasks

- **Descriptive Methods:** Find human-interpretable patterns that describe the data, e.g.
  - Clustering
  - Dimensionality Reduction
  - Association Rule Discovery
  - Sequential Pattern Discovery
- **Predictive Methods:** Use some variables to predict unknown or future values of other variables, e.g.
  - Classification
  - Regression
  - Novelty Detection

Data Mining: Culture

- Data mining overlaps with:
  - Databases: Large-scale data, simple queries
  - Machine learning: Small data, Complex models
  - Statistics: Predictive Models
- Different cultures:
  - To a DB person, data mining is an extreme form of analytic processing – queries that examine large amounts of data
    - Result is the query answer
  - To a stats/ML person, data-mining is the inference of models
    - Result is the parameters of the model



Relation between Data Mining and Data Analytics

- Analytics include both data analysis (mining) and communication (guide decision making)
- Analytics is not so much concerned with individual analyses or analysis steps, but with the entire methodology

Meaningfulness of Answers

- A big data-analytics risk is that you will "discover" patterns that are meaningless
- Statisticians call it Bonferroni's principle:
  - (roughly) if you look in more places for interesting patterns than your amount of data will support, you are bound to find crap

Examples of Bonferroni's Principle

- Total Information Awareness (TIA)
  - In 2002, intend to mine all the data it could find, including credit-card receipts, hotel records, travel data, and many other kinds of information in order to track terrorist activity
  - A big objection was that it was looking for so many vague connections that it was sure to find things that were bogus and thus violate innocents' privacy

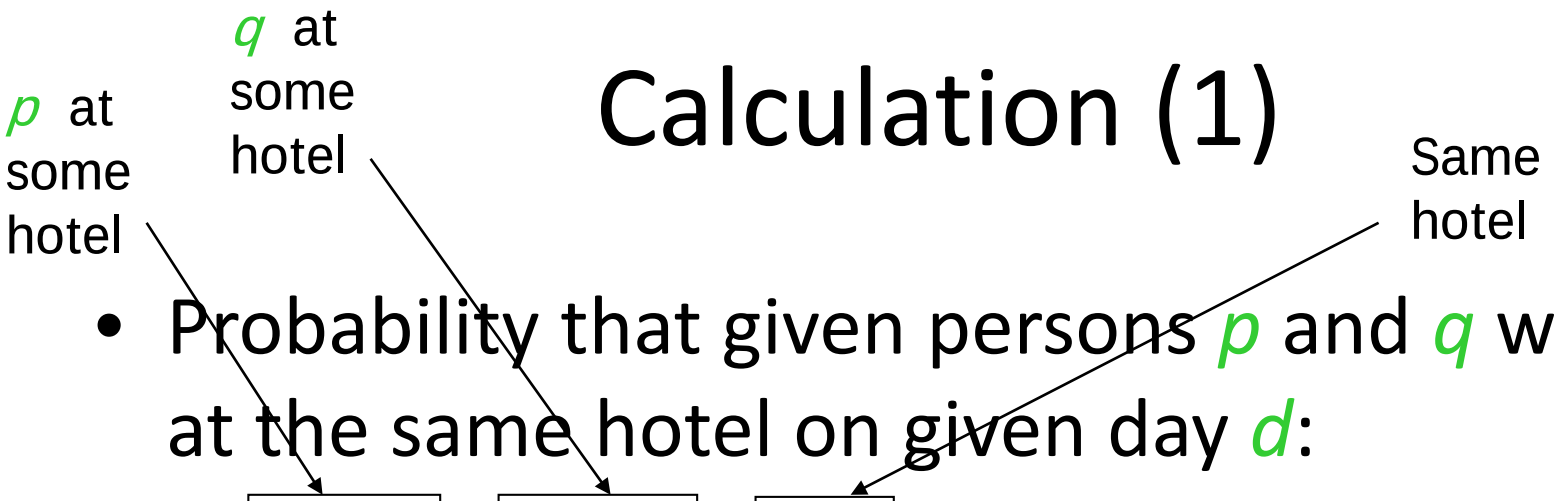
The “TIA” Story

- Suppose we believe that certain groups of evil-doers are meeting occasionally in hotels to plot doing evil
- We want to find (unrelated) people who at least twice have stayed at the same hotel on the same day

Details of The "TIA" Story

- 10^9 people might be evil-doers
- Examining hotel records for 1000 days
- Each person stays in a hotel 1% of the time (10 days out of 1000)
- Hotels hold 100 people (so 10^5 hotels, 1% of total people)
- If everyone behaves randomly (i.e., no evil-doers) will the data mining detect anything suspicious?

Calculation (1)



- Probability that given persons p and q will be at the same hotel on given day d:
  - $\boxed{1/100} \times \boxed{1/100} \times \boxed{10^{-5}} = 10^{-9}$.
- Probability that p and q will be at the same hotel on given days d_1 and d_2:
  - $10^{-9} \times 10^{-9} = 10^{-18}$.
- Pairs of days:
  - 5×10^5

Calculation (2)

- Probability that *p* and *q* will be at the same hotel on some two days:
  - $5\times10^{5} \times 10^{-18} = 5\times10^{-13}$
- Pairs of people:
  - 5×10^{17}
- Expected number of "suspicious" pairs of people:
  - $5\times10^{17} \times 5\times10^{-13} = 250{,}000$

Summary of The "TIA" Story

- Suppose there are 10 pairs of evil-doers who definitely stayed at the same hotel twice
- Analysts have to sift through 250,000 candidates to find the 10 real cases
- Make sure the property, e.g., two people stayed at the same hotel twice, does not allow so many possibilities that random data will surely produce "facts of interest"
- Understanding Bonferroni's Principle will help you look a little less stupid than a parapsychologist

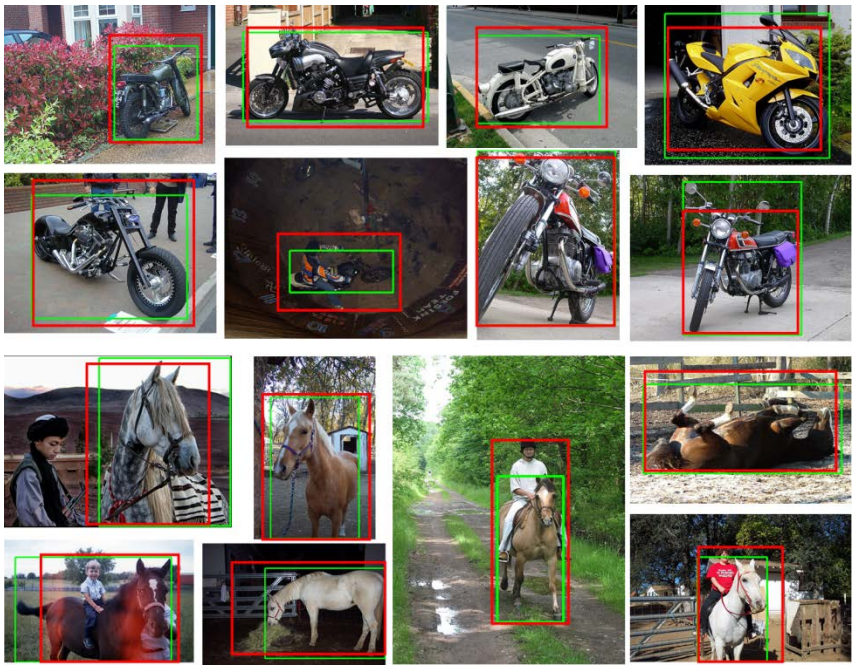
In-class Practice

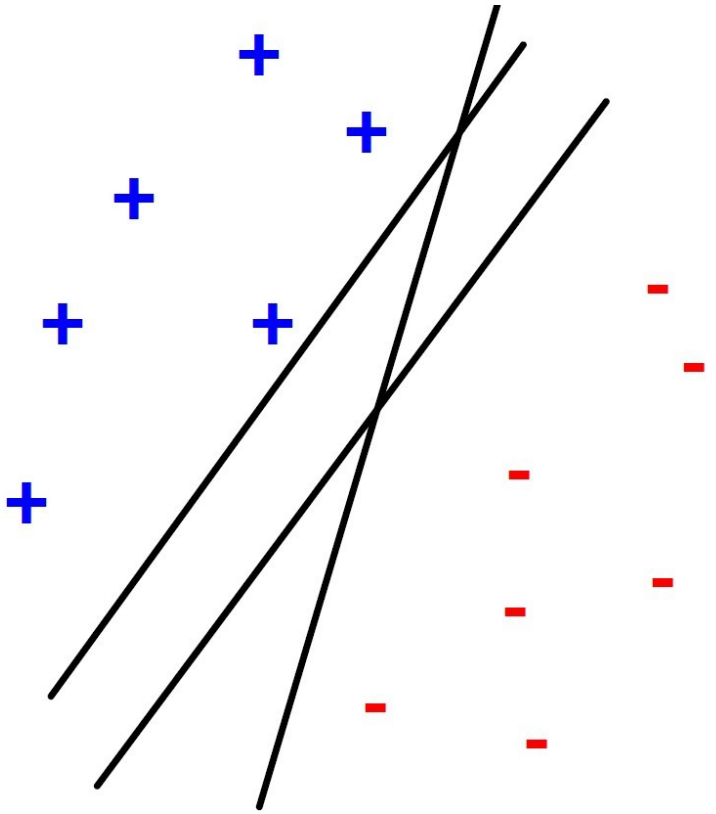
- Go to practice

Seven Typical Statistical Problems

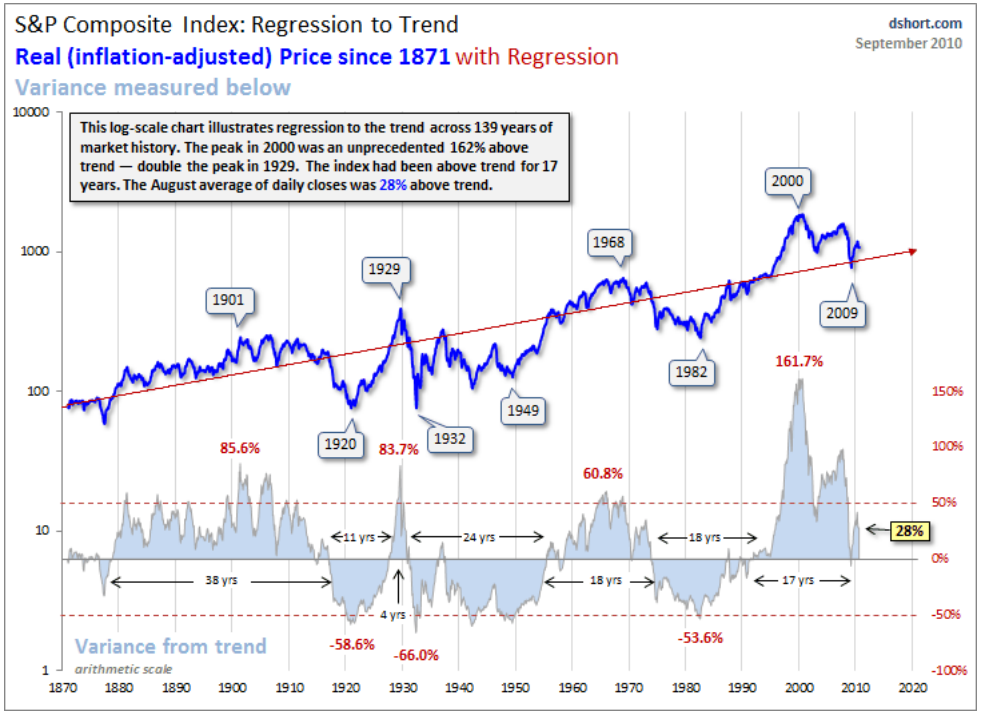
1. Object detection(e.g. quasars): classification
2. Photometric redshift estimation: regression, conditional density estimation
3. Multidimensional object discovery: querying, dimension reduction, density estimation, clustering
4. Point-set comparison: testing and matching
5. Measurement errors: errors in variables
6. Extension to time domain: time series analysis
7. Observation costs: active learning

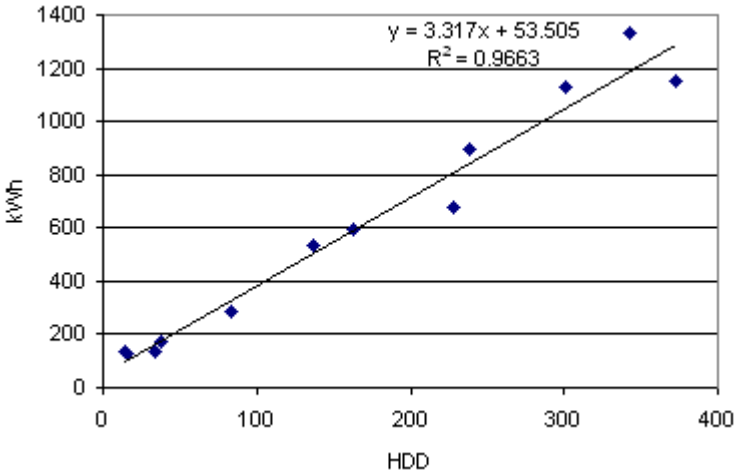
Object Detection: Classification



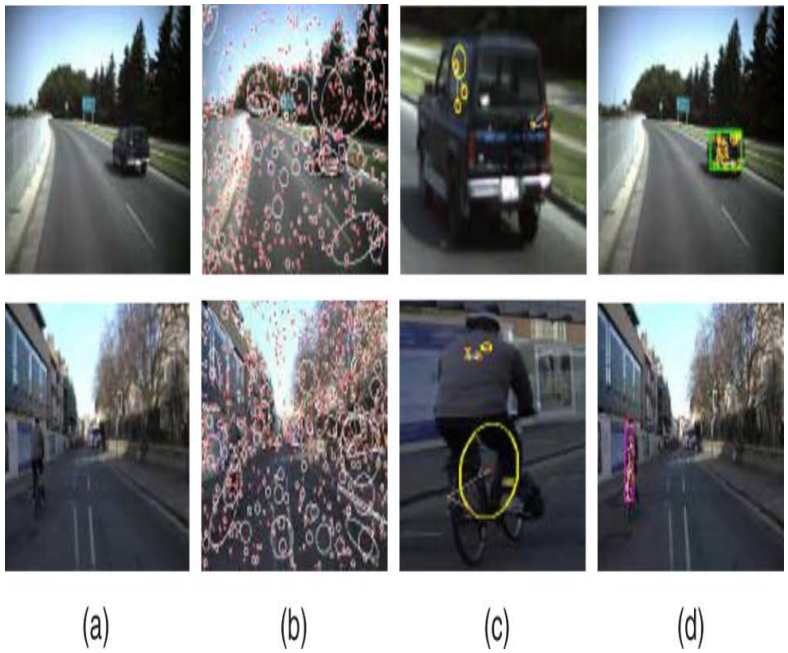


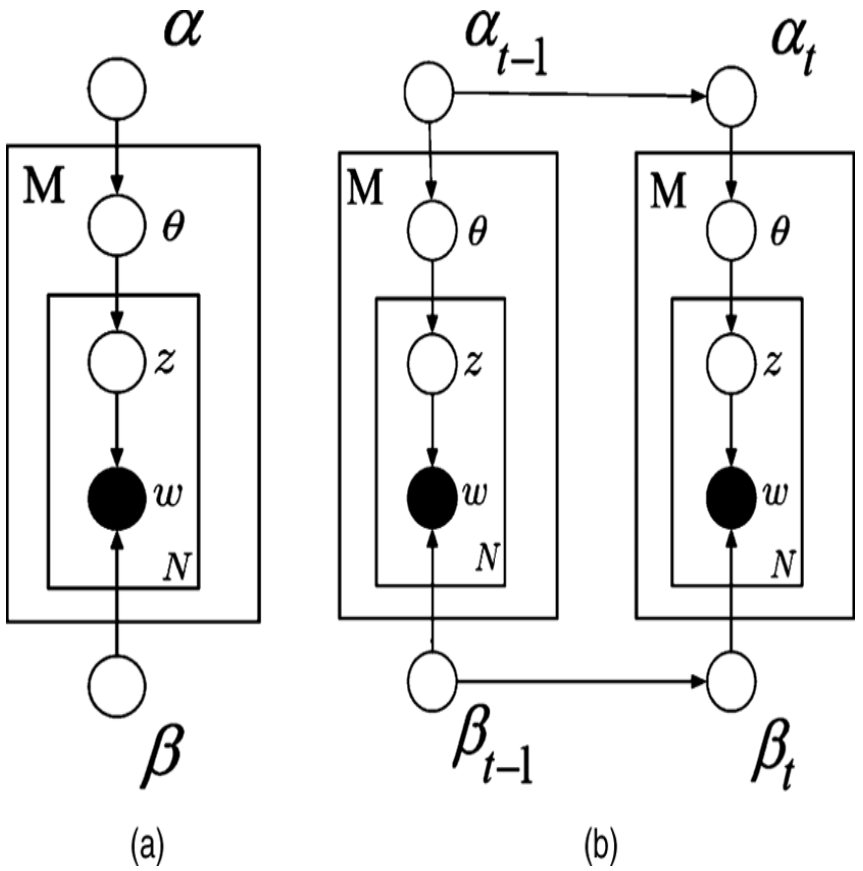
Regression/Conditional Density Estimation



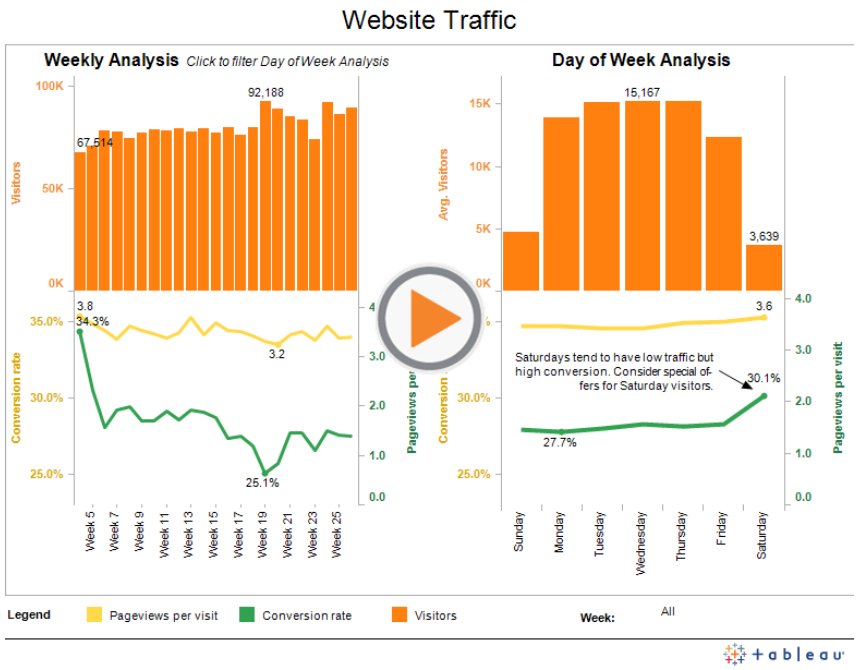


Querying/Dimension Reduction/Density Estimation/Clustering





Time Series Analysis



Seven Lessons in Learning from Big Data

1. Big data is a fundamental phenomenon
2. The system must change
3. Simple solutions run out of steam
4. ML becomes important
5. Data quality becomes important
6. Temporal analysis become important
7. Prioritized sensing becomes important

Seven Lessons in Learning from Big Data

1. Big data is a fundamental phenomenon
2. The system must change
3. Simple solutions run out of steam
4. ML becomes important
5. Data quality becomes important
6. Temporal analysis become important
7. Prioritized sensing becomes important

Seven Lessons in Learning from Big Data

1. Big data is a fundamental phenomenon
2. The system must change
3. Simple solutions run out of steam
4. ML becomes important
5. Data quality becomes important
6. Temporal analysis become important
7. Prioritized sensing becomes important

Current Options

1. Subsample (e.g. then use R, Weka)
2. Use a simpler method (e.g. linear)
3. Use brute force (e.g. Hadoop)
4. Faster algorithm

What Makes this Hard?

1. The key bottlenecks are fundamental computer science/numerical methods problems of many types
2. Useful speedups are needed.
    1. Error guarantees
    2. Known runtime growths

What Makes this Hard?

1. The key bottlenecks are fundamental computer science/numerical methods problems of many types
2. Useful speedups are needed
    1. Error guarantees
    2. Known runtime growths

Seven Lessons in Learning from Big Data

1. Big data is a fundamental phenomenon
2. The system must change
3. Simple solutions run out of steam
4. ML becomes important
5. Data quality becomes important
6. Temporal analysis become important
7. Prioritized sensing becomes important

Seven Lessons in Learning from Big Data

1. Big data is a fundamental phenomenon
2. The system must change
3. Simple solutions run out of steam
4. ML becomes important
5. Data quality becomes important
6. Temporal analysis become important
7. Prioritized sensing becomes important

Seven Lessons in Learning from Big Data

1. Big data is a fundamental phenomenon
2. The system must change
3. Simple solutions run out of steam
4. ML becomes important
5. Data quality becomes important
6. Temporal analysis become important
7. Prioritized sensing becomes important

Seven Typical Tasks of Machine Learning/Data Mining

1. **Querying:** spherical range-search **O(N)**, orthogonal range-search **O(N)**, nearest-neighbor **O(N)**, all-nearest-neighbors **$O(N^2)$**
2. **Density estimation:** mixture of Gaussians, kernel density estimation **$O(N^2)$**, kernel conditional density estimation **$O(N^3)$**
3. **Classification:** decision tree, nearest-neighbor classifier **$O(N^2)$**, kernel discriminant analysis **$O(N^2)$**, support vector machine **$O(N^3)$**, L_p SVM
4. **Regression:** linear regression, LASSO, kernel regression **$O(N^2)$**, Gaussian process regression **$O(N^3)$**
5. **Dimension reduction:** PCA, non-negative matrix factorization, kernel PCA **$O(N^3)$**, maximum variance unfolding **$O(N^3)$**; Gaussian graphical models, discrete graphical models
6. **Clustering:** k-means, mean-shift **$O(N^2)$**, hierarchical (FoF) clustering **$O(N^3)$**
7. **Testing and matching:** MST **$O(N^3)$**, bipartite cross-matching **$O(N^3)$**, n-point correlation 2-sample testing **$O(N^n)$**, kernel embedding

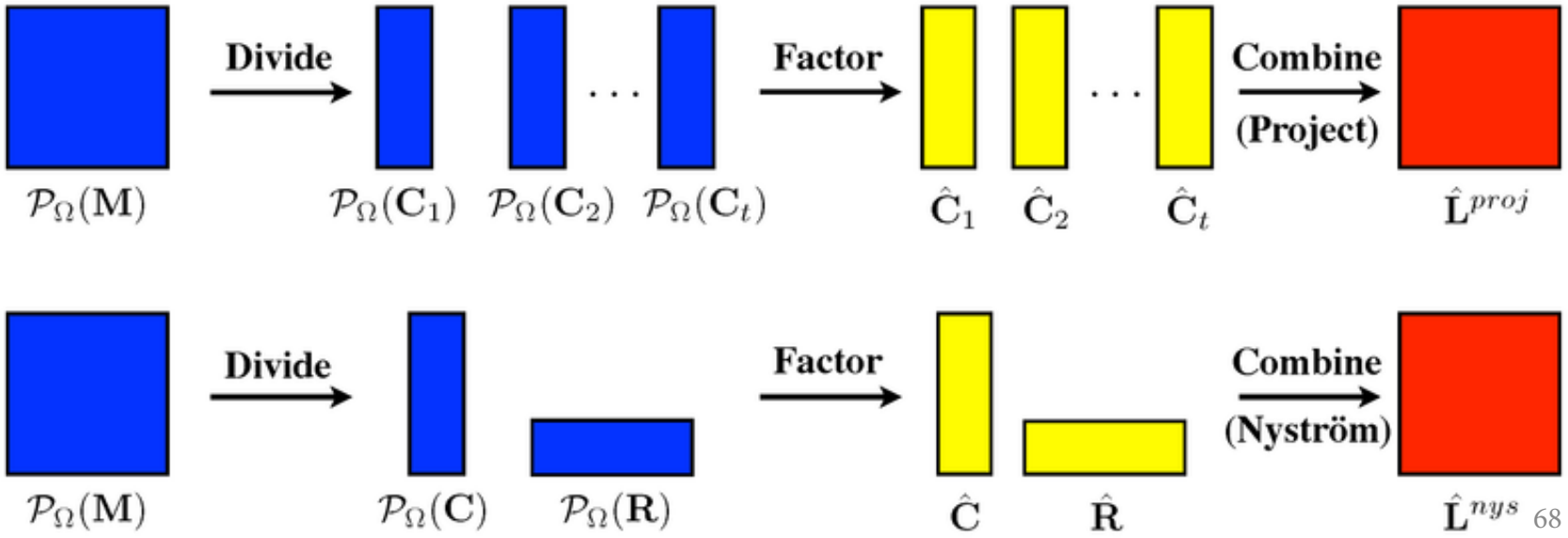
Seven Typical Tasks of Machine Learning/Data Mining

1. **Querying:** spherical range-search **O(N),** orthogonal range-search **O(N),** nearest-neighbor **O(N),** all-nearest-neighbors **O(N^2)**
2. **Density estimation:** mixture of Gaussians, kernel density estimation **O(N^2),** kernel conditional density estimation **O(N^3)**
3. **Classification:** decision tree, nearest-neighb[illegible]lassifier **O(N^2)**, kernel discriminant analysis [illegible], supp[illegible]e **O(N^3)**, L_p SVM
4. **Regressio[illegible]** [illegible]aussian process regression **O(N^3)**
5. **Dimension re[illegible]** [illegible]ix factorization, kernel PCA **O(N^3)**, maximum[illegible]olding **O(N^3)**, Gaussian graphical models, discrete graphical models
6. **Clustering:** k-means, mean-shift **O(N^2),** hierarchical (FoF) clustering **O(N^3)**
7. **Testing and matching:** MST **O(N^3)**, bipartite cross-matching **O(N^3)**, n-point correlation 2-sample testing **O(N^n)**, kernel embedding

Computational Problem!

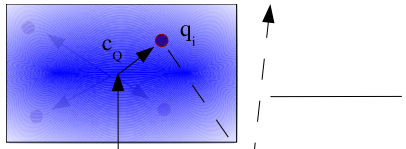
1. Divide and Conquer

- Multidimensional trees:
  - K-d trees [Bentley 1970], ball-trees [Omohundro 1991], spill trees [Liu, Moore, Gray, Yang,nips2004], cover tree [Beygelzimer et al.2006] , cosine tree [Holmes, Isbell, Gray, Nips 2009], subspace trees [Lee and Gray nips 2009], cone trees [Ram and Gray kdd2012], max-margin trees [Ram and Gray SDM 2012], kernel trees [Ram and Gray]

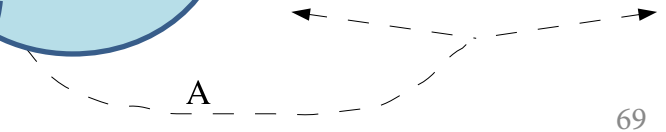


2. Function Transforms

- **Fastest** approach for:
  - **Kernel estimation** (low-ish dimension): *dual-tree fast Gauss transforms* (multipole/Hermite expansions) [Lee, Gray, Ma NIPS 2005], [Le
  - **KDE and G** estimati regres *Fourie* in prep]



Generalized N-body approach is fundamental: like multidimensional generalization of FFT



3. Sampling

- **Fastest** approach for (approximate):

– **PCA:** *cosine trees* [Holmes, Gray, Isbell, NIPS 2008]

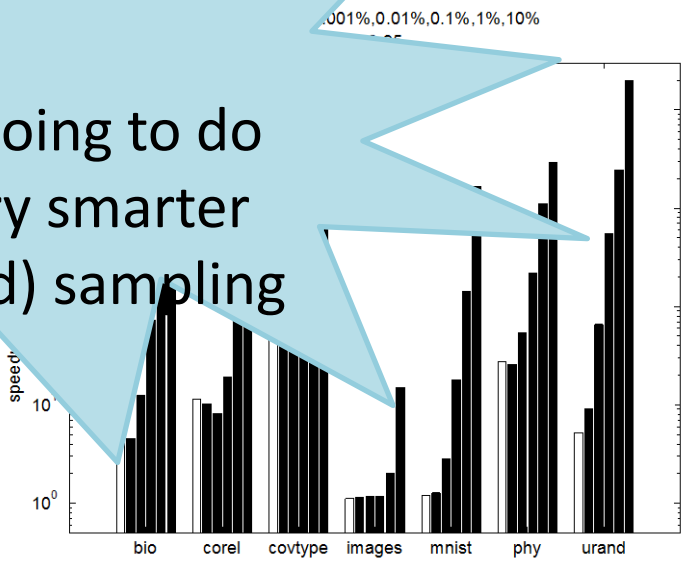
– **Kernel estimation:** bandwidth learning [Holnes, Gray, Isbell, NIPS 2006],[Holmes, Gray, Isbell, UAI 2007], *Monte Carlo multipole method (with SVD trees)* [Lee & Gray, NIPS 2009], *shadow densities* [Kingravi et al., under review]

–**Nearest-neighbo**: distance-appro *spill tree* h random proj[Liu, Moore, Gray, Yang, NIPS 200*k-appro*: [ivang, Gray, NIPS 2009]

Rank-approximate

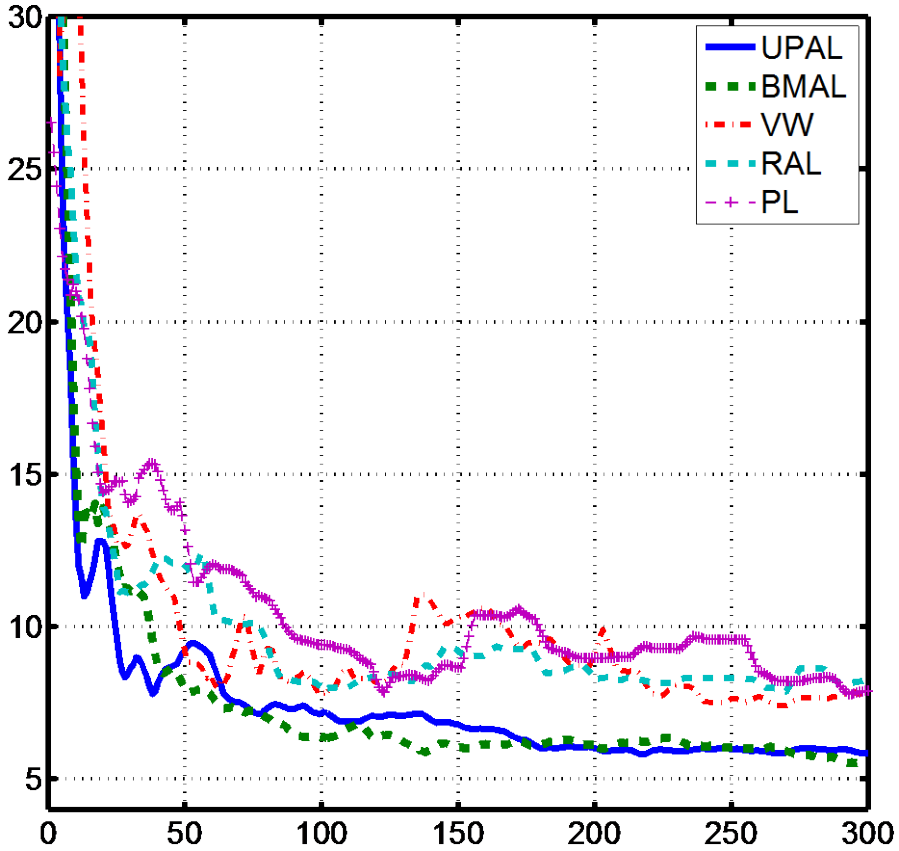
- Best meaning-retain approximation c face of high-dimensional
- More accurate than LSH

3. If you're going to do sampling, try smarter (e.g. stratified) sampling



3. Sampling

- ***Active learning*: the sampling can depend on previous samples**
  - **Linear classifiers:** rigorous framework for *pool-based active learning* [Sastry and Gray, AISTATS 2012]
    - Empirically allows reduction in the number of objects that require labeling
    - Theoretical rigor: unbiasedness



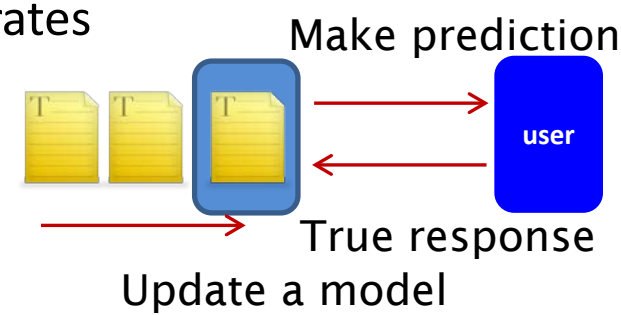
4. Caching

- **Fastest** approach for (using disk):
  - **Nearest-neighbor, 2-point:** *Disk-based tree algorithms* in Microsoft SQL Server [Riegel, Aditya, Budavari, Gray, in prep]
    - Builds k-d tree on top of built-in B-trees
    - Fixed-pass algorithm to build k-d tree

No. of points	MLDB(Dual tree)	Naive
40,000	8 seconds	159 seconds
200,000	43 seconds	3480 seconds
10,000,000	297 seconds	80 hours
20,000,000	29 mins 27 sec	74 days
40,000,000	58 mins 48 sec	280 days
40,000,000	112 mins 32 sec	2 years

5. Streaming/Online

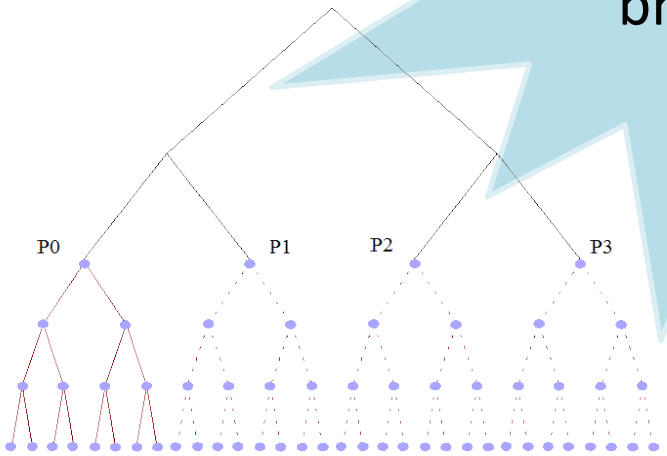
- **Fastest** approach for (approximate, or streaming):
  - *Online learning/stochastic optimization*: just use the current sample to update the gradient
    - **SVM** (squared hinge loss): stochastic Frank-Wolfe[Ouyang and Gray, SDM 2010]
    - **SVM, LASSO, et al.:** noise-adaptive stochastic approximation (NASA)[Ouyang and Gray, KDD 2010], accelerated non-smooth SGD (ANSGD) [Ouyang and Gray, ICML 2012]
      - faster than SGD
      - solves step size problem
      - beats all existing convergence rates

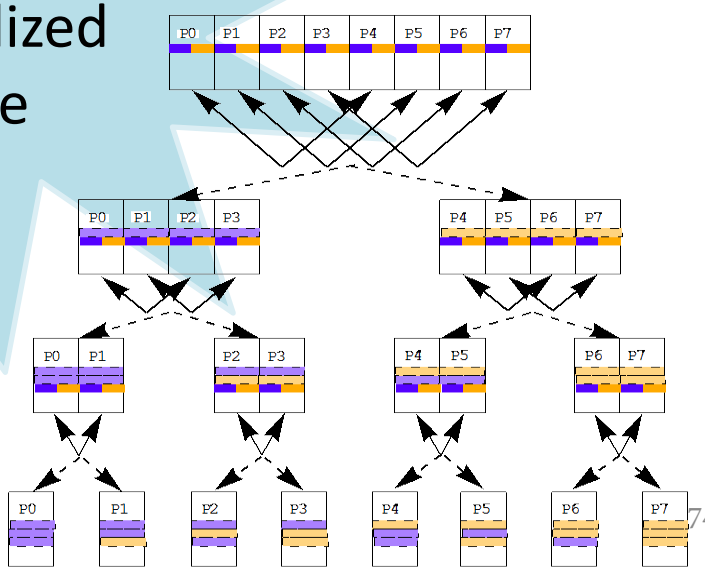


6. Parallelism

- **Fastest** approach for (using many machines):
  - **KDE, GP, n-point:** *distributed trees* [Lee and Gray , SDM 2012 Best Paper], 6000+ cores; [March et al, Supercomputing 2012], 100K cores
    - Each process owns the global tree and its local tree
    - First log p levels built in parallel; each process determines where to send data
- Asynchronous averaging approval [illegible]
  - **SVM, LASSO, et al.** [illegible] [Quyang and Gray, in prep]
    - Provable [illegible]

6. Parallelized fast alg. > parallelized brute force





7. Transformations between Problems

- Change the problem type:
  - Linear algebra on kernel matrices → N-body inside conjugate gradient [Gray, TR 2004]
  - Euclidean graphs → N-body problems [March & Gray, KDD 2010]
  - HMM as graph → matrix factorization [Tran & Gray, in prep]
- Optimizations: reformulate the objective and constraints:
  - Maximum variance unfolding: *SDP via Burer-Monteiro convex relaxation* [Vasiloglou, Gray, Anderson MLSP 2009]
  - L_q SVM, $0<q<1$: *DC programming* [Guan & Gray, CSDA 2-11]
  - L_0 SVM: mixed integer nonlinear program via perspective cuts [Guan & Gray, under review]
  - Do reformulations automatically [Agarwal et al, PADL 2010],[Bhat et al, POPL 2012]

7. Transformations between Problems

- Create new ML methods with desired computational properties:
  - Density estimation trees: nonparametric density estimation, O(NlogN) [Ram & Gray, KDD 2011]
  - Local linear SVMs: nonli… O(NlogN) [Sastry &…
  - Discriminative local coo… classification O(NlogN) […er review]

When all else fails, change the problem


What is cloud computing?

The best thing since sliced bread?

- Before clouds…
  - Grids
  - Vector supercomputers
  - …
- Cloud computing means many different things:
  - Large-data processing
  - Rebranding of web 2.0
  - Utility computing
  - Everything as a service

Rebranding of web 2.0

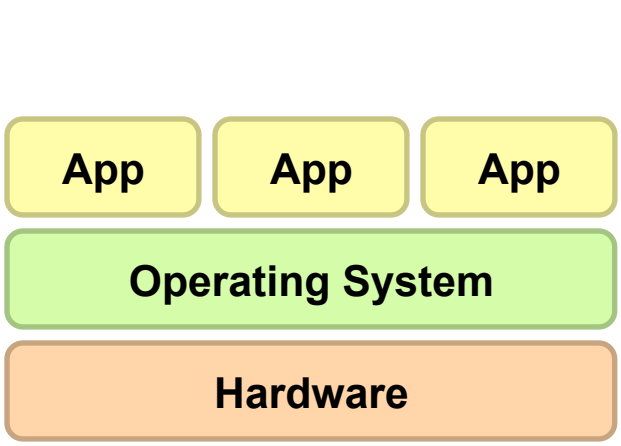
- Rich, interactive web applications
  - Clouds refer to the servers that run them
  - AJAX as the de facto standard (for better or worse)
  - Examples: Facebook, YouTube, Gmail, …
- "The network is the computer": take two
  - User data is stored "in the clouds"
  - Rise of the netbook, smartphones, etc.
  - Browser *is* the OS

Utility Computing

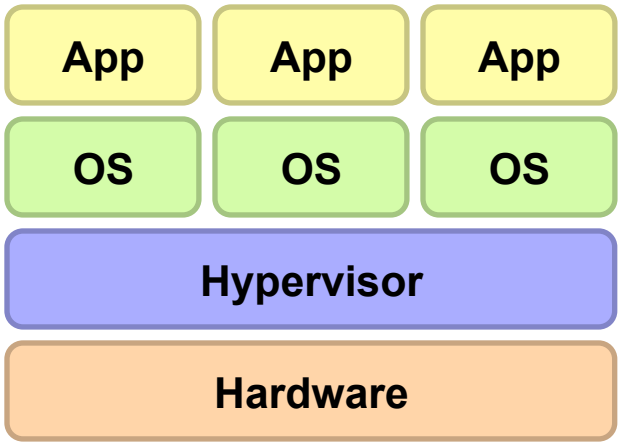
- What?
  - Computing resources as a metered service (“pay as you go”)
  - Ability to dynamically provision virtual machines
- Why?
  - Cost: capital vs. operating expenses
  - Scalability: “infinite” capacity
  - Elasticity: scale up or down on demand
- Does it make sense?
  - Benefits to cloud users
  - Business case for cloud providers

I think there is a world market for about five computers.
– Thomas J Watson of IBM, 1943

Enabling Technology: Virtualization



Traditional Stack



Virtualized Stack

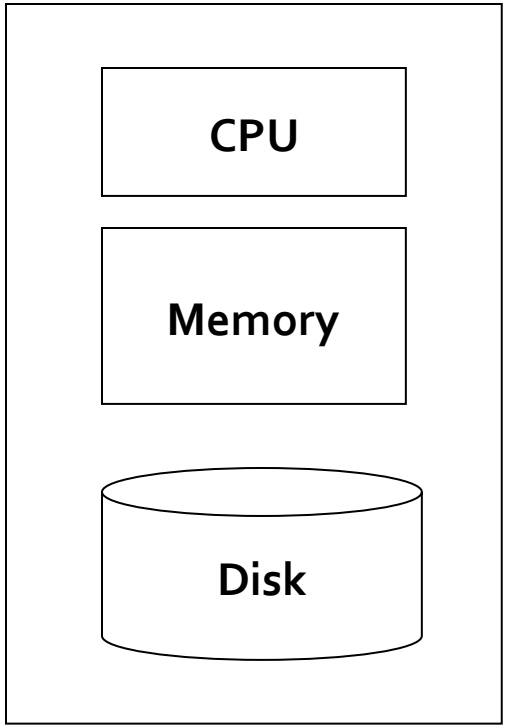
Everything as a Service

- Utility computing = Infrastructure as a Service (IaaS)
  - Why buy machines when you can rent cycles?
  - Examples: Amazon's EC2, Rackspace
- Platform as a Service (PaaS)
  - Give me nice API and take care of the maintenance, upgrades, …
  - Example: Google App Engine
- Software as a Service (SaaS)
  - Just run it for me!
  - Example: Gmail, Salesforce

How do we scale up processing for Big Data ?

Or: How to run Algorithms on MANY REAL and FAULTY boxes ?

Single Node Architecture

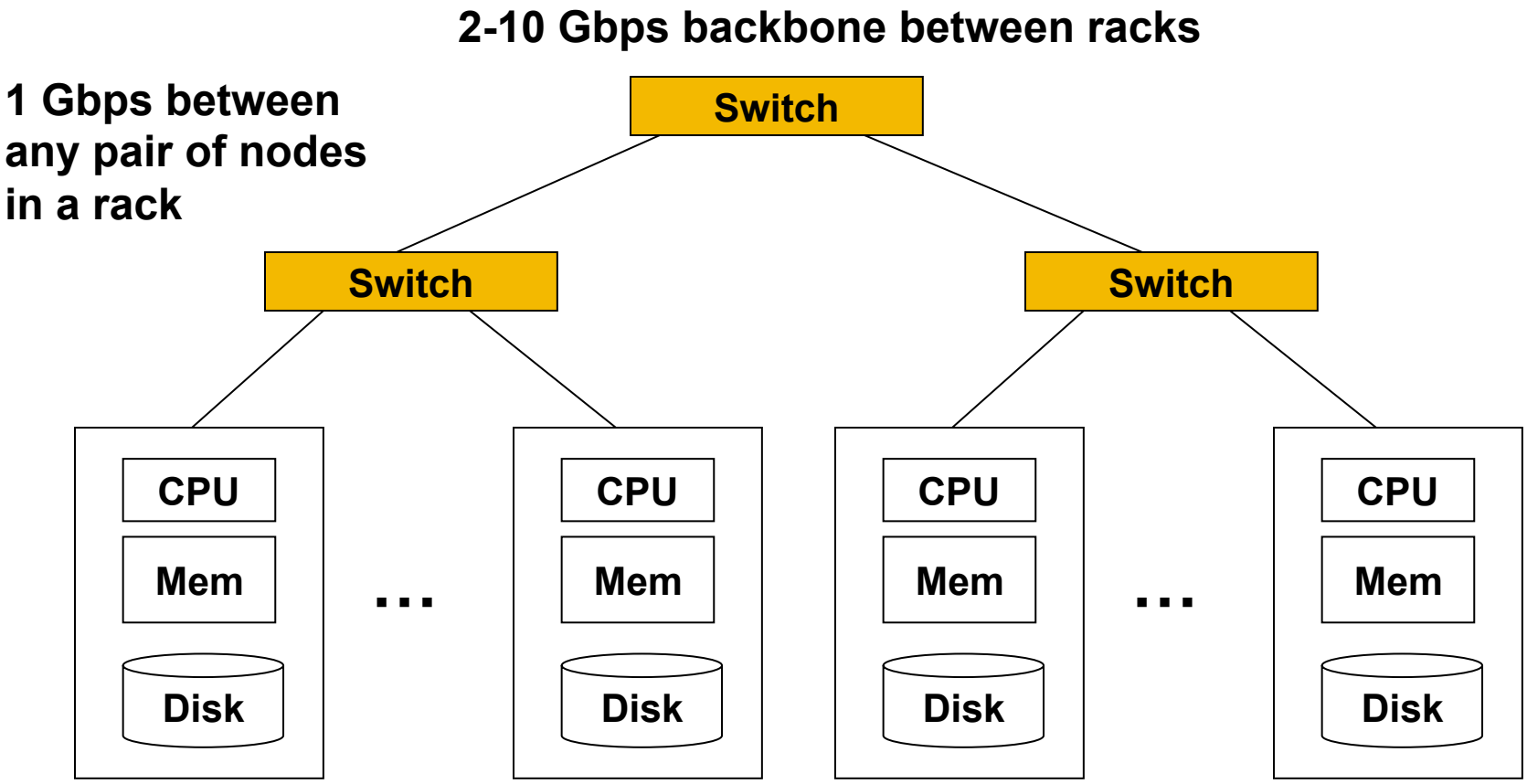


"Classical"
Machine Learning, Statistics,
Data Mining

Motivation: Google Example

- 20+ billion web pages x 20KB = 400+ TB
- 1 computer reads 30-35 MB/sec from disk
  - ~4 months to read the web
- ~1,000 hard drives to store the web
- Takes even more to **do** something useful with the data!
- **Today, a standard architecture for such problems is emerging:**
  - Cluster of commodity Linux nodes
  - Commodity network (ethernet) to connect them

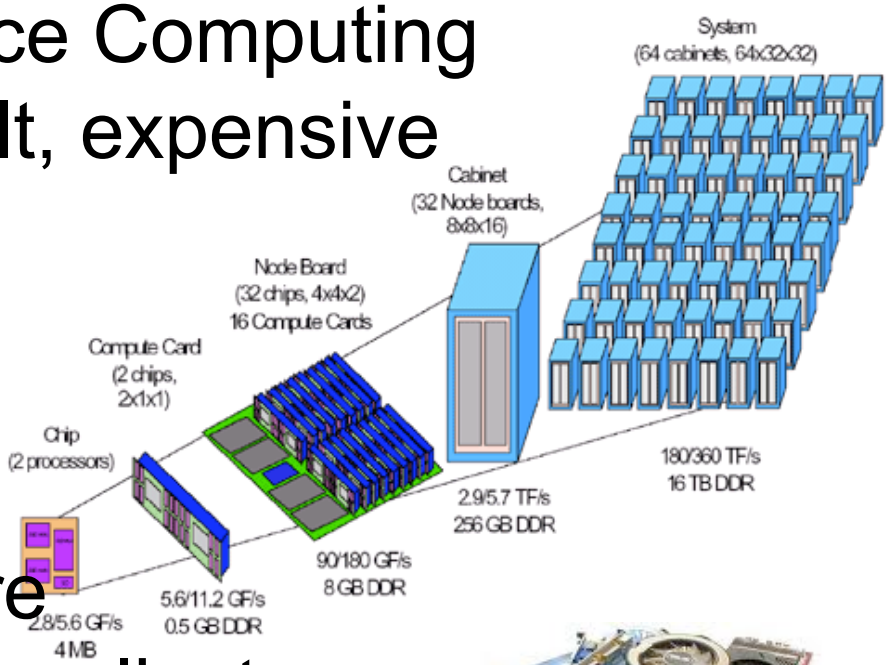
Cluster Architecture



In 2011, it was guestimated that Google had 1M machines, http://bit.ly/Shh0RO

Using Commodity Hardware

- 80-90's: High Performance Computing
  Very reliable, custom built, expensive



- Now: Consumer hardware
  Cheap, efficient, easy to replicate,
  BUT not very reliable,
- MUST deal with it!

Why commodity machines?

	HP INTEGRITY SUPERDOME-ITANIUM2	HP PROLIANT ML350 G5
Processor	64 sockets, 128 cores (dual-threaded), 1.6 GHz Itanium2, 12 MB last-level cache	1 socket, quad-core, 2.66 GHz X5355 CPU, 8 MB last-level cache
Memory	2,048 GB	24 GB
Disk storage	320,974 GB, 7,056 drives	3,961 GB, 105 drives
TPC-C price/performance	$2.93/tpmC	$0.73/tpmC
price/performance (server HW only)	$1.28/transactions per minute	$0.10/transactions per minute
Price/performance (server HW only) (no discounts)	$2.39/transactions per minute	$0.12/transactions per minute

Source: Barroso and Urs Hölzle (2009); performance figures from late 2007

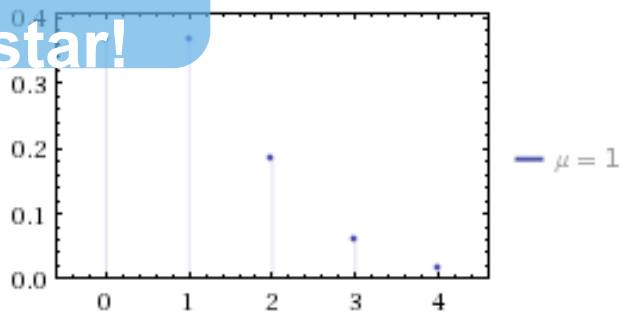
Fault Tolerance

- Performance goal
  - 1 failure per year
  - for a 1000-machine Cluster
- Poisson approximation

$$\Pr(n) = \frac{1}{n!} e^{-\mu} \mu^n$$

  - Assume failure rate μ per machine
  - Poisson rates of independent random variables are additive, so we can combine

not IBM Deskstar!

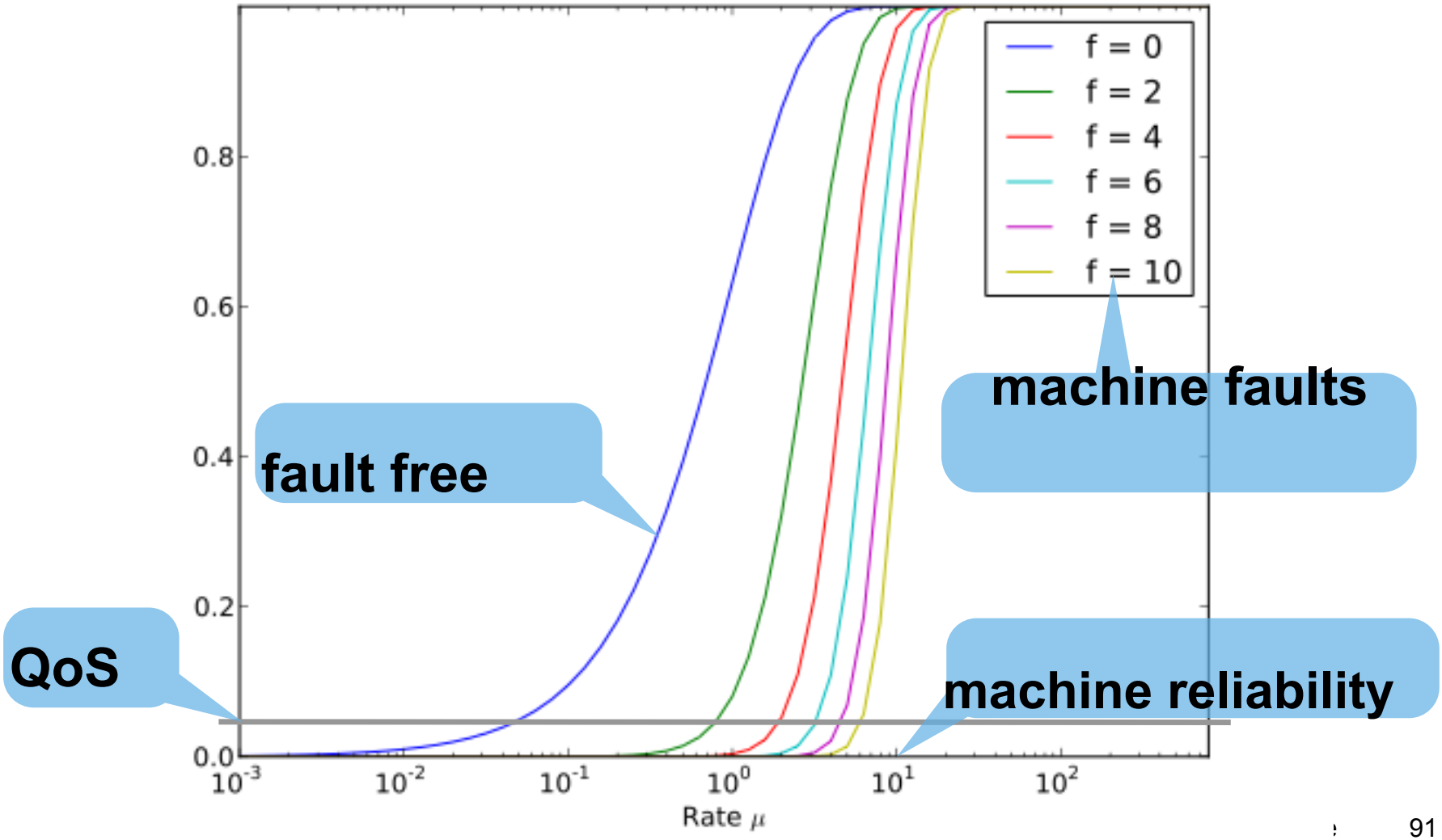


=> With Fault Intolerant Engineering
We need a rate of 1 failure per 1000 years per machine

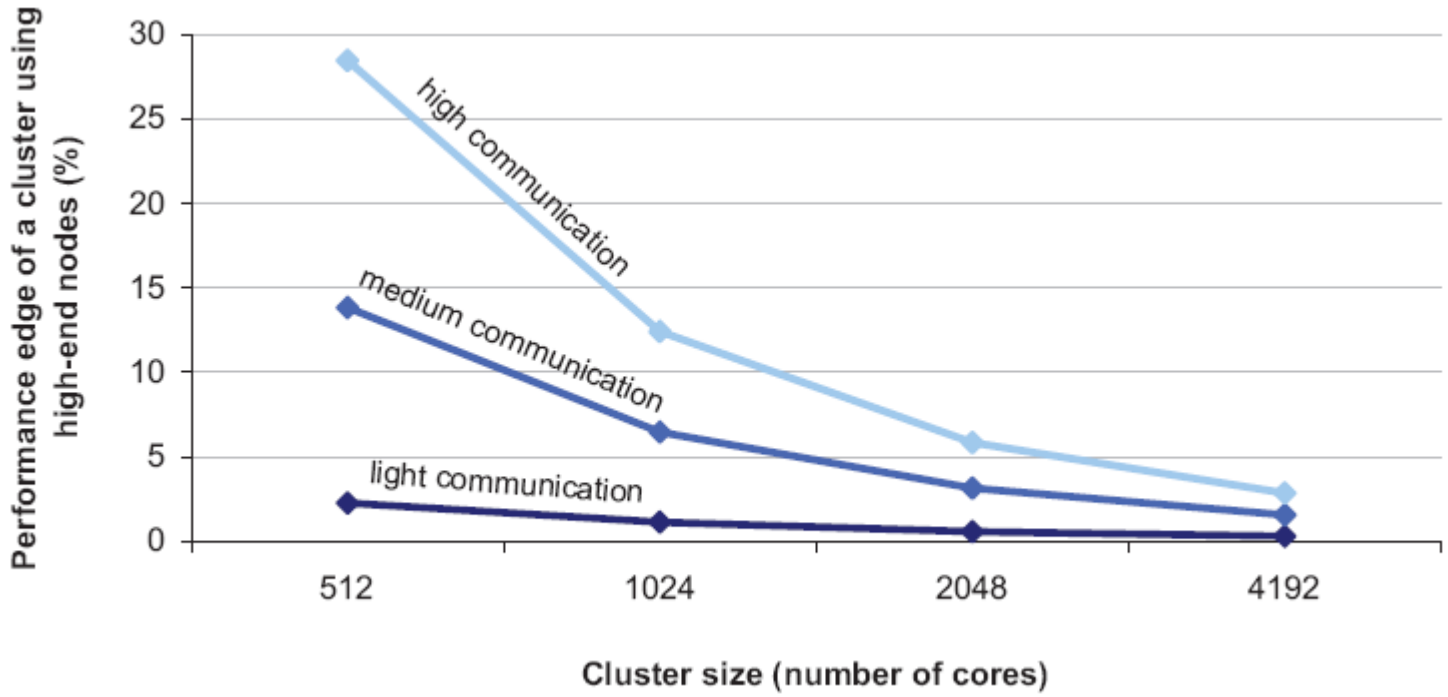
- Fault tolerance
  Assume we can tolerate k faults among m machines in t time

$$\Pr(f > k) = 1 - \sum_{n=0}^{k} \frac{1}{n!} e^{-\lambda t} (\lambda t)^n$$

Fault tolerance



Advantages of scaling “out”



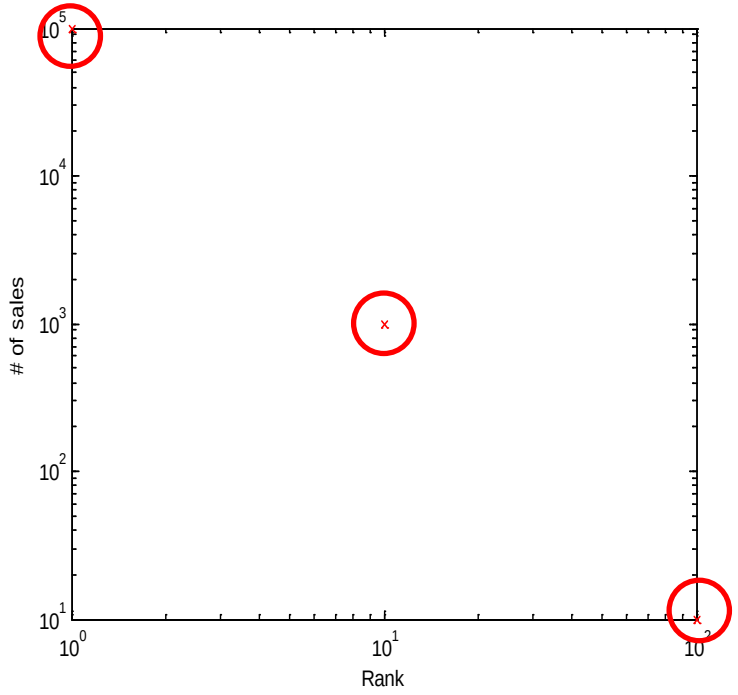
So why not?

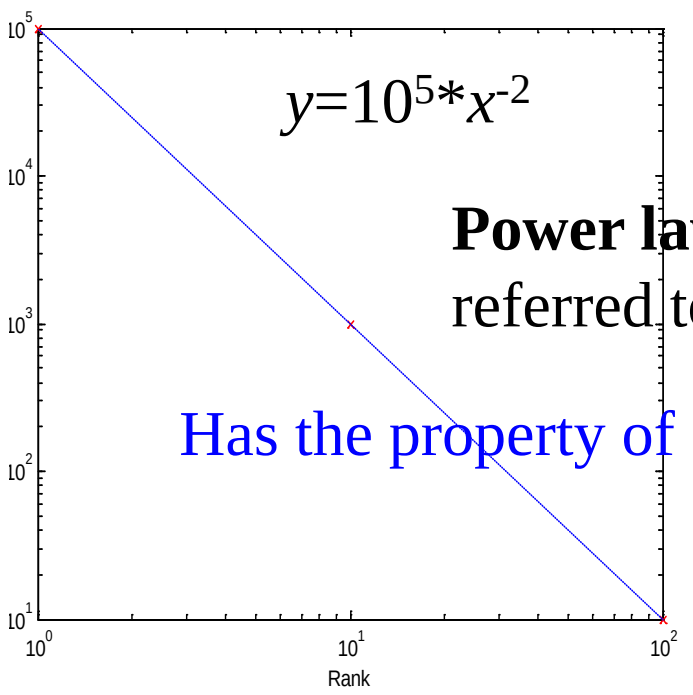
In-class Practice

- Let us examine fragrance sales at ebay in a year. Suppose
  - the best selling product sold 100,000 pieces,
  - the 10^{th} best-selling product sold 1,000 pieces,
  - the 100^{th} best selling product sold 10 pieces.
- How to derive the relationship between the number of fragrance sold and the order?

In-class Practice

- Let y be the number of sales of the x-th best-selling fragrance products in a year at ebay.





Power law: also referred to Zipf's law

Has the property of scale invariance

Go back